

WHAT IS CLAIMED IS:

1. An agent system for processing information by an agent, the agent system being connected to another agent system of a different type through a network, comprising:

wrapper class memory configured to store component data of wrappers corresponding to each type of agent system, said wrappers being for processing information in the corresponding agent system; and

wrapper generator configured to generate a wrapper corresponding to an agent system of a different type based on the component data of the different type stored in said wrapper class memory;

wherein said wrapper acts for an agent to use resources in the agent system of a different type.

2. The agent system according to claim 1, further comprising:

same type node data memory configured to store node data of a same type agent system and the agent system of a different type discriminately;

different type node data memory configured to store access data for a node included in the agent system of a different type; and

correspondence table configured to store correspondence data between a category of movement command

and a type of other agent system.

3. The agent system according to claim 2,
further comprising:

movement request detector configured to detect a movement command to move to other node from a script of the agent, for deciding whether the other node is a different type agent system by referring to said same type node data memory, and for extracting the type of other agent system corresponding to the movement command by referring to said correspondence table if the other node is a different type agent system.

4. The agent system according to claim 3,
further comprising:

retrieval unit configured to retrieve the component data of a wrapper corresponding to the type of other agent system extracted by said movement request detector from said wrapper class memory,

wherein said wrapper generator extracts the access data of the other node as a movement destination node from said different type node data memory after generating said wrapper, supplies the access data to the wrapper through a common interface, and activates said wrapper.

5. The agent system according to claim 4,

wherein said wrapper executes action through an external communication unit in the different type agent system, and returns to an original node as a movement source node in the same type agent system according to a movement command to the original node in the script of the agent.

6. The agent system according to claim 5,
wherein said wrapper comprises:

activation unit configured to activate a wrapped agent whose component is based on component of the wrapper in the different type agent system;

different type movement memory configured to store data of moving the agent in the different type agent system; and

movement source return memory configured to store data of returning the agent from the different type agent system to the movement source agent system.

7. The agent system according to claim 6,
wherein said wrapped agent comprises:

agent state memory configured to store data representing state of the agent in the movement source agent system;

movement unit, generated according to data in said different type movement memory in said wrapper, configured

to store data of moving the wrapped agent in the different type agent system;

return unit, generated according to data in said movement source return memory in said wrapper, configured to store data of returning the agent from the different type agent system to the movement source agent system; and controller configured to execute action of the wrapped agent according to the data in said agent state memory, said movement unit and said return unit.

8. The agent system according to claim 7, wherein if the action of the wrapped agent fails in the different type agent system,

said return unit indicates the agent in the movement source agent system to execute the same action.

9. The agent system according to claim 5, wherein said wrapper comprises:

activation data generator configured to generate activation data of processing to be executed in the different type agent system according to the script of the agent;

activation unit configured to activate another agent in the different type agent system according to the activation data; and

result acquisition unit configured to receive the

processing result executed by another agent in the different type agent system.

10. The agent system according to claim 9, wherein if a processing instruction not executable in the different type agent system is included in the script of the agent,

said activation data generator generates activation data to temporarily return the agent to the movement source agent system, to execute the processing instruction in the movement source agent system and to resend the processing result to the movement destination agent system.

11. The agent system according to claim 5, wherein said wrapper comprises:

inherent processing handler configured to execute an inherent processing previously registered in the different type agent system; and

function call unit configured to call a function of the different type agent system to execute the inherent processing through a function call interface of the destination node in the different type agent system.

12. The agent system according to claim 11, wherein if the instruction in the script of agent is a command to execute the inherent processing,

the source node in the original agent system calls the function of the different type agent system by said inherent processing handler through said function call unit and said function call interface.

13. A method for processing information by an agent in an agent system, the agent system being connected to another agent system of a different type through a network, comprising the steps of:

storing component data of wrappers corresponding to each type of agent system, the wrapper being for processing information in the corresponding agent system; and

generating a wrapper corresponding to an agent system of a different type based on the stored component data of the different type;

wherein the wrapper acts for an agent to use resources in the agent system of a different type.

14. The method according to claim 13, further comprising the steps of:

discriminately storing node data of a same type agent system and the agent system of a different type;

storing access data for node included in the agent system of a different type; and

storing correspondence data between a category of movement command and a type of other agent system.

15. The method according to claim 14,
further comprising the steps of:
detecting a movement command to move to other node
from a script of the agent:
deciding whether the other node is a different type
agent system by referring to the stored node data; and
extracting the type of other agent system
corresponding to the movement command by referring to the
stored correspondence data if the other node is a different
type agent system.

16. The method according to claim 15,
further comprising the steps of:
retrieving the component data of a wrapper
corresponding to the type of other agent system from the
stored component data;
extracting the access data of the other node as a
movement destination node from the stored access data after
generating the wrapper;
supplying the access data to the wrapper through a
common interface; and
activating the wrapper.

17. The method according to claim 16,
in the wrapper, further comprising the steps of:

executing action through an external communication unit in the different type agent system; and

returning to an original node as a movement source node in the same type agent system according to a movement command to the original node in the script of the agent.

18. The method according to claim 17,
in the wrapper, further comprising the steps of:
activating a wrapped agent whose component is based on component of the wrapper in the different type agent system;

storing movement data of the agent in the different type agent system; and

storing return data of the agent from the different type agent system to the movement source agent system.

19. The method according to claim 18,
in the wrapped agent, comprising the steps of:
storing state data of the agent in the movement source agent system;

storing data of moving the wrapped agent in the different type agent system according to the movement data;

storing data of returning the agent from the different type agent system to the movement source agent system according to the return data; and

executing action of the wrapped agent according to the

stored data.

20. The method according to claim 19,
further comprising the step of:
if the action of the wrapped agent fails in the
different type agent system,
indicating the agent in the movement source agent
system to execute the same action.

21. The method according to claim 17,
in the wrapper, comprising the steps of:
generating activation data of processing to be
executed in the different type agent system according to
the script of the agent;
activating another agent in the different type agent
system according to the activation data; and
receiving the processing result executed by another
agent in the different type agent system.

22. The method according to claim 21,
further comprising the step of:
if a processing instruction not executable in the
different type agent system is included in the script of
the agent,
generating activation data to temporarily return the
agent to the movement source agent system, to execute the

processing instruction in the movement source agent system and to resend the processing result to the movement destination agent system.

23. The method according to claim 17, further comprising the steps of: in the wrapper, executing an inherent processing previously registered in the different type agent system; and calling a function of the different type agent system to execute the inherent processing through a function call interface of the destination node in the different type agent system.

24. The method according to claim 23, further comprising the step of :
if the instruction in the script of agent is a command to execute the inherent processing,
in the source node of the original agent system, comprising the step of :
calling the function of the different type agent system to execute the inherent processing through the wrapper and the function call interface.

25. A computer readable memory containing computer readable instructions to process information by an agent in an agent system, the agent system being connected to

another agent system of a different type through a network, comprising:

the first instruction to cause a computer to store component data of wrappers corresponding to each type of agent system, the wrappers being for processing information in a corresponding agent system; and

the second instruction to cause a computer to generate a wrapper corresponding to an agent system of a different type based on the stored component data of the different type stored;

wherein the wrapper acts for an agent to use resources in the agent system of a different type.